

An Overview of Text Games and World Modeling

Peter Jansen

Associate Professor @ University of Arizona
Visiting Scientist @ Allen Institute for AI (AI2)

cognitiveai.org

<http://textgames.org>

Peter Jansen

Associate Professor @ University of Arizona

Research Interests (AI+NLP)

- Inference, reasoning, explanations
- World modeling, simulation
- Scientific reasoning

Background (not all who wander are lost):

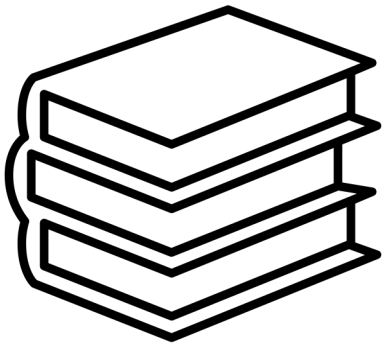
- CS, Physics, Cognitive Psychology, Neuroscience, ECE

Fun Facts:

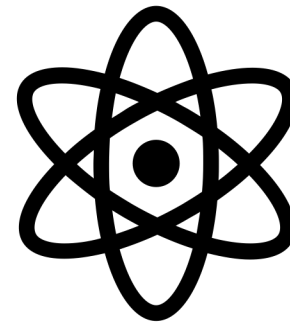
- I've written/helped write a lot of the modern simulators for text games in the last 2 years
- I've been a guest at Star Trek conventions for being the scientist that made real-life star trek technology (tricorder)



Today



Overview of
Text Game Research



ScienceWorld: Is your Agent
Smarter than a 5th grader?

Learning Objectives

- Leave with a high-level overview of the broad categories of research in text games
- Understand basic modeling formalisms
- Find one or more papers (or, broad areas) that you find exciting
- ScienceWorld: Understand that text games offer a new, interactive way of testing an agent's knowledge of the world (sometimes with surprising results)

A Systematic Survey of Text Worlds as Embodied Natural Language Environments

Peter Jansen

School of Information, University of Arizona, USA

pajansen@email.arizona.edu

Abstract

Text Worlds are virtual environments for embodied agents that, unlike 2D or 3D environments, are rendered exclusively using textual descriptions. These environments offer an alternative to higher-fidelity 3D environments due to their low barrier to entry, providing the ability to study semantics, compositional inference, and other high-level tasks with rich high-level action spaces while controlling for perceptual input. This systematic survey outlines recent developments in tooling, environments, and agent modeling for Text Worlds, while examining recent trends in knowledge graphs, common sense reasoning, transfer learning of Text World performance to higher-fidelity environments, as well as near-term development targets that, once achieved, make Text Worlds an attractive general research paradigm for natural language processing.

1 Introduction

Embodied agents offer an experimental paradigm to study the development and use of semantic representations for a variety of real-world tasks, from household tasks (Shridhar et al., 2020a) to navigation (Guss et al., 2019) to chemical synthesis (Tamari et al., 2021). While robotic agents are a primary vehicle for studying embodiment (e.g. Cangelosi and Schlesinger, 2015), robotic models are costly to construct, and experiments can be slow or difficult to scale. Virtual agents and embodied virtual environments help mitigate many of these issues, allowing large-scale simulations to be run in parallel orders of magnitude faster than real world environments (e.g. Deitke et al., 2020), while controlled virtual environments can be constructed for exploring specific tasks – though this benefit

Zork

North of House

You are facing the north side of a white house. There is no door here, and all the windows are barred.

>go north

Forest

This is a dimly lit forest, with large trees all around. One particularly large tree with some low branches stands here.

>climb large tree

Up a Tree

You are about 10 feet above the ground nestled among some large branches. On the branch is a small birds nest. In the bird's nest is a large egg encrusted with precious jewels, apparently scavenged somewhere by a childless song-bird.

>take egg

Taken.

>climb down tree

Forest

>go north

Forest

This is a dimly lit forest, with large trees all around. To the east, there appears to be sunlight.

>go east

Clearing

You are in a clearing, with a forest surrounding you on the west and south. There is a pile of leaves on the ground.

>move leaves

Done. A grating appears on the ground.

>open grating

The grating is locked.

Table 1: An example Text World interactive fiction environment, Zork (Lebling et al., 1979), frequently used as a benchmark for agent performance. User-entered actions are *italicized*.

emerged as a recent methodological focus that allow studying many embodied research questions while reducing some of the development costs associated with modeling complex and photorealistic 3D environments (e.g. Côté et al., 2018). More than simply reducing development costs, Text Worlds also offer paradigms to study developmental knowledge representation, embodied task learning, and

Interactive Virtual Environments

A list of research articles published on interactive text-based virtual environments, agents, and related areas.

New to text-based interactive virtual environments?

[Learn More](#)

Table of Contents

- [Simulators: Text-game Engines](#)
- [Environments: Specific Interactive Games/Environments/Benchmarks](#)
- [World Generation: Automatic World Generation](#)
- [Agents: Agents/Agent Architectures](#)
- [Data: Data/Resources](#)
- [Position Papers](#)
- [Shared Tasks](#)
- [Social Agents: Agent-user or agent-agent dialog](#)
- [Surveys](#)
- [Other](#)

Legend: ★ [Very recently added](#) ★ [Added in last 90 days](#) ★ [Added in last year](#)

<http://textgames.org>

● Simulators

Papers that describe simulators, which are like game engines that specific games/environments can be created in.

★ [ByteSized32: A Corpus and Challenge Task for Generating Task-Specific World Models Expressed as Text Games](#)

Ruoyao Wang, Graham Todd, Eric Yuan, Ziang Xiao, Marc-Alexandre Côté, Peter Jansen — EMNLP 2023

TL DR: Presents ByteSized32, a corpus of 32 text games expressed as approximately 1000 lines of Python each. Shows that a

What do people write papers about?

Category	# of Papers
Simulators	7
Environments	18
World Generation	7
Agents	49
Social Agents	5
Data	5
Position Papers	2
Shared Tasks	5
Surveys	2

What do people write papers about?

Category		# of Papers
Simulators	5 papers	7
Environments	5 papers	18
World Generation		7
Agents	5 papers	49
Social Agents		5
Data		5
Position Papers		2
Shared Tasks		5
Surveys		2

Simulators

Z-Machine/ZIL (1980s)

- Formalism and interpreted language created by Infocom in the 1980s
 - Interpreted language for portability
- Allowed programmers to “easily” specify games, without having to worry much about implementing the text parser
- Resembles LISP
- Underlying choices in world modeling (e.g. how objects are represented) still used today

3.1 What an Object Definition Looks Like

Here's what the definition of Zork I's brass lantern looks like:

```
<OBJECT LANTERN
  (LOC LIVING-ROOM)
  (SYNONYM LAMP LANTERN LIGHT)
  (ADJECTIVE BRASS)
  (DESC "brass lantern")
  (FLAGS TAKEBIT LIGHTBIT)
  (ACTION LANTERN-F)
  (FDESC "A battery-powered lantern is on the trophy
case.")
  (LDESC "There is a brass lantern (battery-powered)
here.")
  (SIZE 15)>
```

2.1 What a Typical Room Definition Looks Like

Here's what the definition of the Living Room from Zork I looks like:

```
<ROOM LIVING-ROOM
  (LOC ROOMS)
  (DESC "Living Room")
  (EAST TO KITCHEN)
  (WEST TO STRANGE-PASSAGE IF CYCLOPS-FLED ELSE
    "The wooden door is nailed shut.")
  (DOWN PER TRAP-DOOR-EXIT)
  (ACTION LIVING ROOM-F)
  (FLAGS RLANDBIT ONBIT SACREDBIT)
  (GLOBAL STAIRS)
  (THINGS <> NAILS NAILS-PSEUDO)>
```

The “Object Tree”

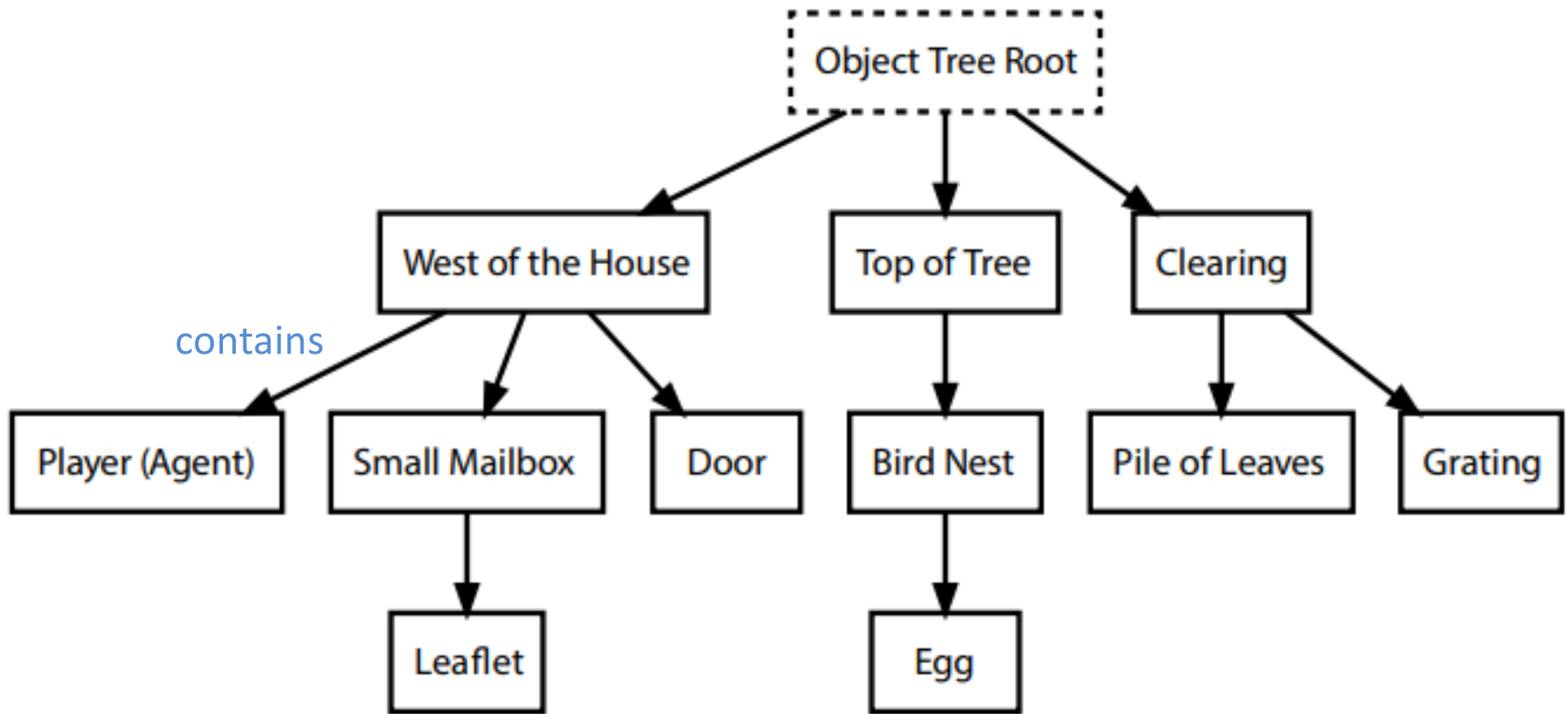


Figure 1: An example partial object tree from the interactive fiction game *Zork* (Lebling et al., 1979).

Inform7: Programming Interactive Fiction in Natural Language (2006)

Inform7

The Kitchen is a room. "A well-stocked Kitchen."

The Living Room is north of the Kitchen.

A stove is in the Kitchen. A table is in the Kitchen. A plate is on the table.

An Apple is on the plate. The Apple is edible.

The cook is a person in the Kitchen. The description is "A busy cook."

The ingredient list is carried by the cook. The description is "A list of ingredients for the cook's favorite recipe."

Instead of listening to the cook:

say "The cook asks if you can help find some ingredients, and hands you a shopping list from their pocket.";

move the ingredient list to the player.

Environment Interpreter

Kitchen

A well-stocked Kitchen.

You can see a stove, a table (on which is a plate (on which is an Apple)) and a cook here.

>eat apple

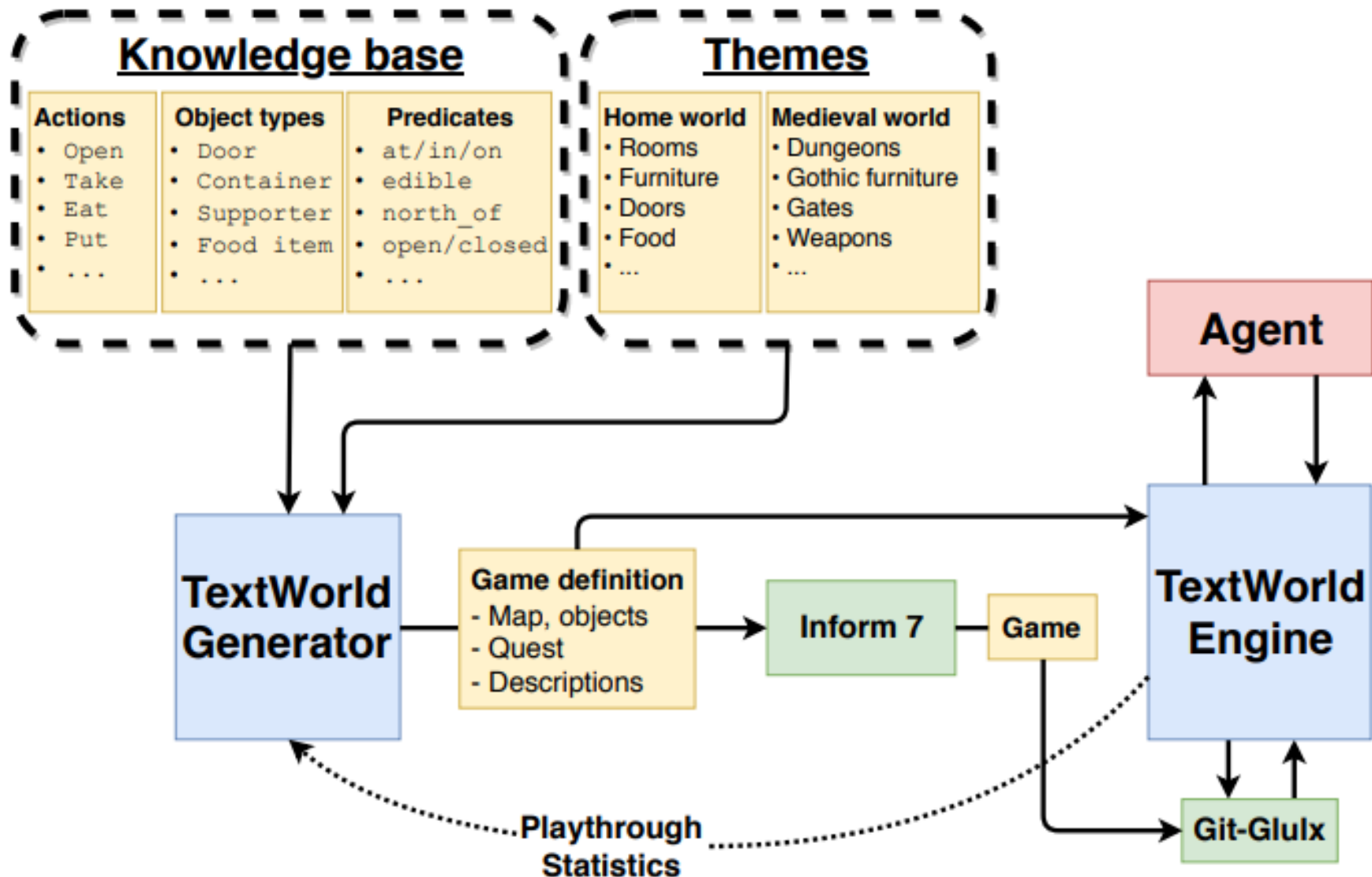
(first taking the Apple)

You eat the Apple. Not bad.

>listen to cook

The cook asks if you can help find some ingredients, and hands you a shopping list from their pocket.

TextWorld: A Learning Environment for Text-Based Games (Cote et al., 2018)



ScienceWorld (Wang et al., 2022)

- First new simulator for NLP research (no ZIL!)
- High fidelity
 - Simulation engines for thermodynamics, chemistry, electrical circuits, friction, genetic reproduction, etc.
- Paired with 30 tasks centered around elementary science
- Written as ~30k lines of Scala, with Python interface (**pip** installable)



TextWorldExpress (Jansen and Cote, 2023)

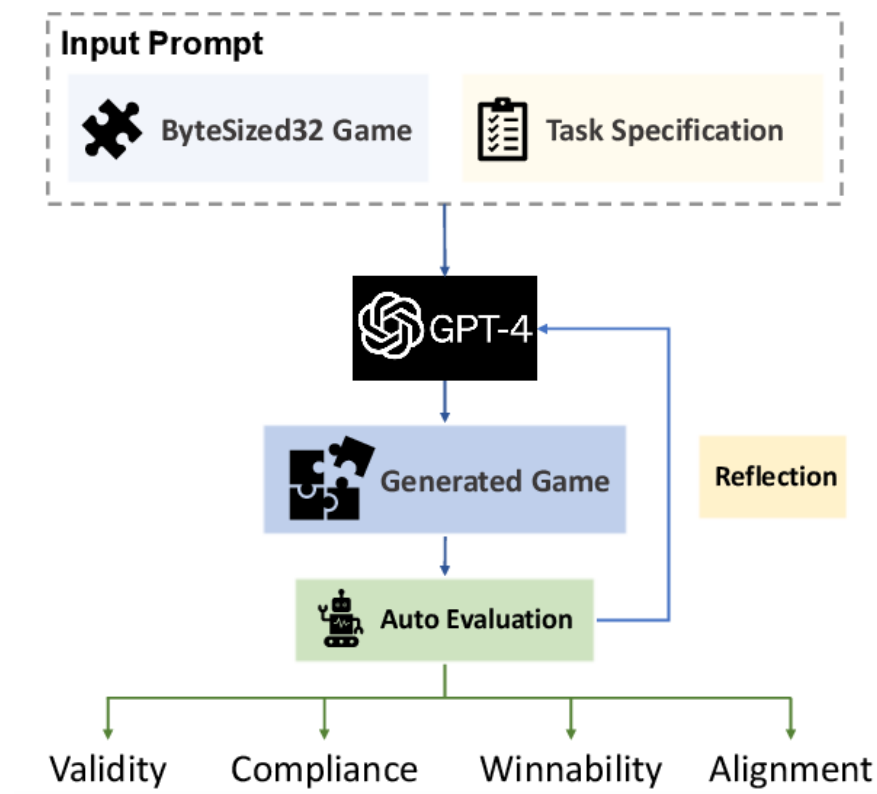
- Text game simulators tend to be shockingly slow (~1 step per second)
- TextWorldExpress can get you to 1 million steps per second!

Environment Simulator	SPS
<i>2D/3D Simulators³</i>	
AI2THOR (Kolve et al., 2017)	30†
MINERL (Guss et al., 2019)	180†
BABYAI (Chevalier-Boisvert et al., 2019)	3k
NETHACK (Küttler et al., 2020)	14k
MEGVERSE (Petrenko et al., 2021)	327k†
<i>Text Game Simulators⁴</i>	
TEXTWORLD (Côté et al., 2018)	300
JERICH0 (Hausknecht et al., 2020)	1
SCIENCEWORLD (Wang et al., 2022)	20
TEXTWORLDEXPRESS (<i>online</i> , PYTHON)	32k
TEXTWORLDEXPRESS (<i>precrawled</i> , PYTHON)	316k
TEXTWORLDEXPRESS (<i>online</i> , JAVA)	212k
TEXTWORLDEXPRESS (<i>precrawled</i> , JAVA)	4M

1-300 steps/sec
vs
30k-4M steps/sec

ByteSized32: Code Generation for Creating New Simulators (Wang et al., 2023)

- Writing simulators takes a lot of time and effort.
- Can we just get an LLM to write a simulator for us, in Python?
- Answer: Kind of. Using a 1-shot prompt, generates runnable games, but often with lots of problems.



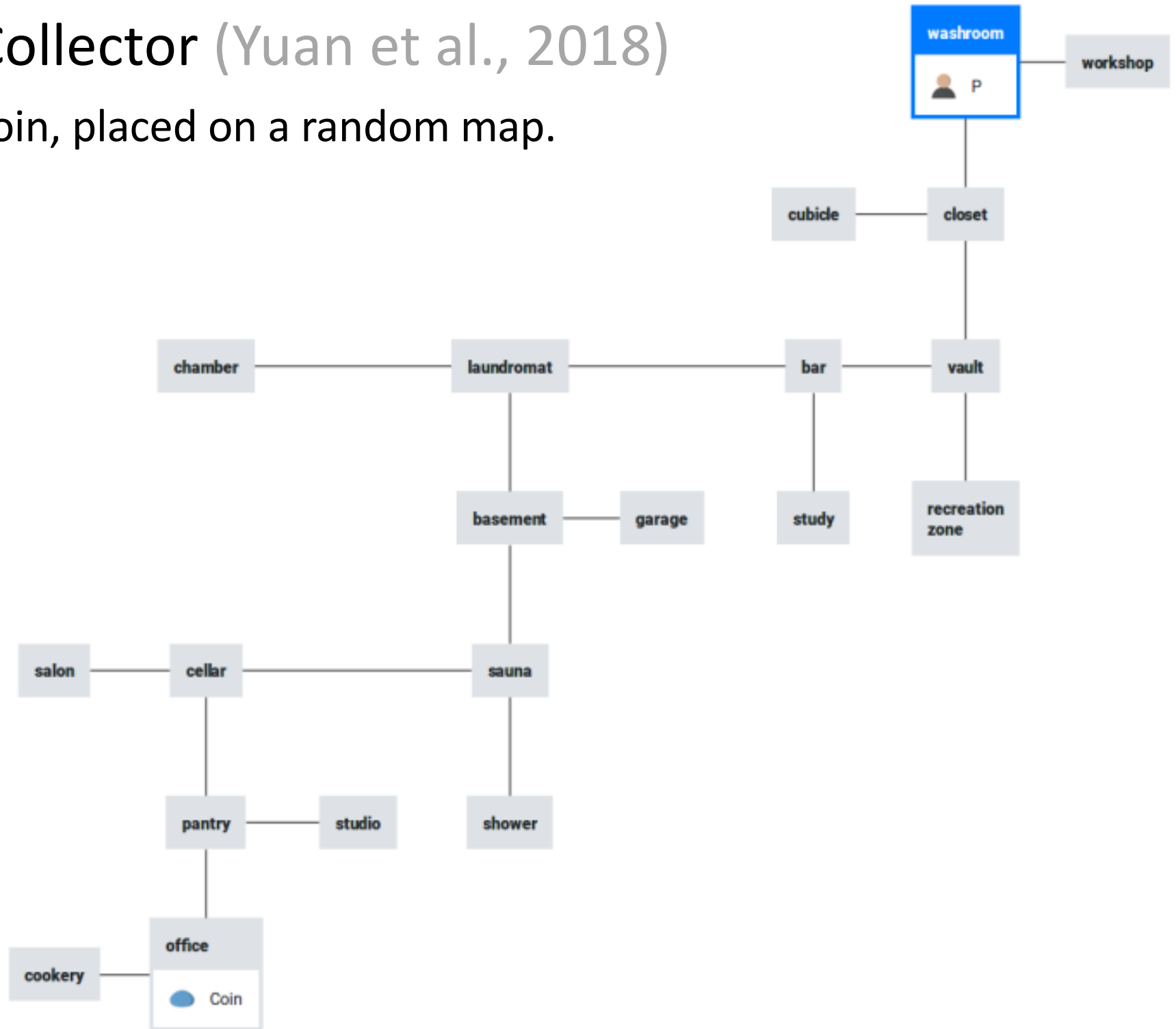
Open Areas of Simulator Research

- Tooling: Make it easy to make a new text game!
- Code generation: Make language models better at generating simulators on their own!
 - Fine tuned models vs n-shot models
 - Iteratively building worlds one piece at a time
 - Correcting their own bugs (reflection)
- LLMs as simulators? (e.g. AI Dungeon)
 - Can we just use an LLM as a simulator, and talk to it?

Environments

Coin Collector (Yuan et al., 2018)

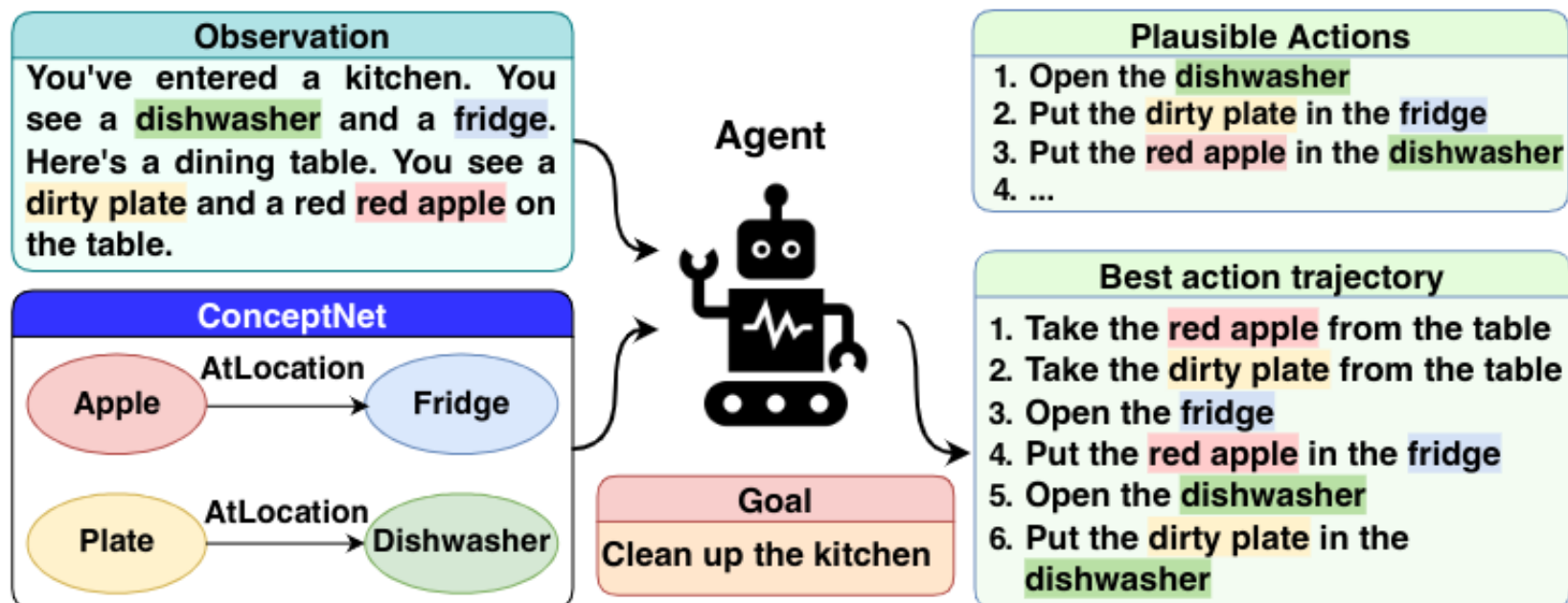
Find a coin, placed on a random map.



TextWorld CommonSense (TWC)

(Murugesan et al., 2021)

- Clean-up game.
 - Place objects (e.g. dirty socks) into their canonical locations (e.g. washing machine)
- Includes a minimal navigation task (up to 3 rooms)
- Agents can solve this, most of the battle is over efficiency.
- Has an implementation in *TextWorldExpress*



CookingWorld (Cote et al., 2020)

- Player is given a recipe
- Must **gather** ingredients (typically ~3) from home environment
- Must **prepare** ingredients (e.g. slice, dice, chop, fry, boil, barbeque) according to the recipe
- Put everything together to make the meal
- Very common benchmark
- Can be surprisingly hard for agents. Need *navigation, reading, pick-and-place, multi-step instruction following*
- Included in TextWorldExpress

CookingWorld Game

North of House

You are in the kitchen. In one part of the room you see a fridge that is closed. There is also a dining chair, that has nothing on it. You also see a dishwasher that is closed. In another part of the room you see a trash can that is closed. In one part of the room you see an oven. There is also a cutlery drawer that is closed. You also see a stove. In another part of the room you see a counter that has a **cookbook**, a green apple, and a red hot pepper on it. In one part of the room you see a kitchen cupboard that is closed. To the South you see the backyard. To the West you see the pantry.

> *read cookbook*

Gather all following ingredients and follow the directions to prepare this tasty meal.

Ingredients: **purple potato**, **yellow bell pepper**.

Directions: dice the **purple potato**, fry the **purple potato**, roast the **yellow bell pepper**, prepare meal.

> *open cutlery drawer*

You open the cutlery drawer. The cutlery drawer contains a **knife**.

> *take knife*

You take the knife.

> *move south*

You are in the backyard. In one part of the room you see a barbeque. There is also a patio chair, that has nothing on it. You also see a clothes line, that has nothing on it. In another part of the room you see a garden that has a raw **purple potato**, and a **yellow bell pepper** on it. In one part of the room you see a workbench, that has nothing on it. There is also a patio table, that has nothing on it. To the North you see the kitchen.

> *take purple potato*

You take the **purple potato**.

> *take yellow bell pepper*

You take the **yellow bell pepper**.

> *move north*

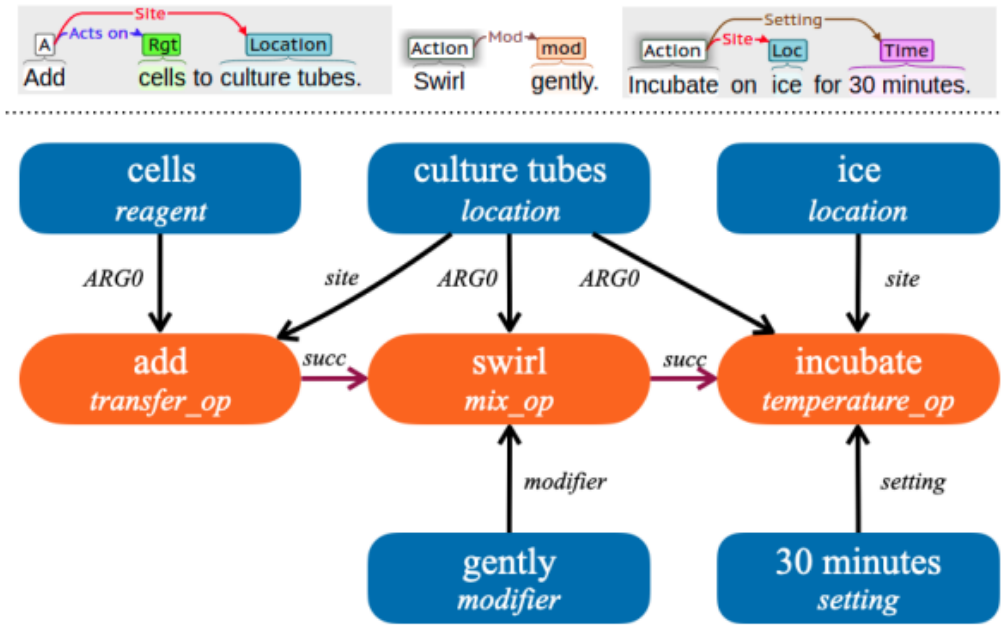
Jericho Benchmark (Hausknecht et al., 2020)



- Benchmark of ~30 common interactive fiction games (e.g. Zork)
- Built a huge amount of infrastructure to support running NLP experiments on these classic games:
 - Valid action detection (e.g. is “*take sword*” a valid action this turn?)
 - Score detection
 - OpenAI Gym-like interface (common with reinforcement learning agents)
 - Baseline agents, and measurements of their performance on all ~30 games.
- Performance of the best agent models is still extremely low on many games (e.g. Zork!)



Biology Wet-lab Experiments with PEG

(Tamari et al., 2021)

- Detailed simulation of real wet-lab experiments, written in TextWorld
- Many domain-specific actions (spin, mix, change temperature, transfer, wash, etc.)
- One of the few examples of a scientific domain being converted to a text game.



 
transfer(cells, culture_tubes)
mix(tubes, 20rpm, 30min)
incubate(tubes, 20f, 30min)

 
transfer(cells, culture_tubes)
mix(tubes, 10rpm, 40min)
incubate(tubes, 10f, 30min)

ALFWorld (Shridhar, 2021)

- Creates text game versions of the 3D Ask-for-ALFRED tasks
 - 6 Pick-and-place tasks
 - e.g. pick-heat-place
- ALFRED is a set of 3D home environments for robotics research implemented in AI2 Thor.
- *Transfer learning*: Shows pretraining on the text environment helps improve performance on a 3D agent!



Figure 1: ALFWorld: Interactive aligned text and embodied worlds. An example with high-level text actions (left) and low-level physical actions (right).

Open Areas of Benchmark Research

- **Long-horizon problems:** Problems that require many (dozens++) of steps to solve
- **Specific Reasoning:** Benchmarks that categorize the kinds of common-sense reasoning required to solve a task (e.g. does an agent know how to read a map? Does it know how to build a campfire?)
 - Lots of research in augmenting LLMs with symbolic modules to augment their reasoning
- **Irregular Tasks:** Environments that require very different action sequences across different tasks (so that an agent can't easily distill the recipe for solving a task from a few examples).
 - **Ablations:** e.g. in ScienceWorld, we randomly set the stove to be broken. Can an agent figure out how to boil water without the stove? (e.g. chop down a tree and make a camp fire?)

Agents

Nearly every agent is terrible at Interactive Fiction games

Model	<i>Detective (E)</i>	<i>Zork1 (M)</i>	<i>Zork3 (M)</i>	<i>OmniQuest (M)</i>	<i>Spirit (H)</i>	<i>Enchanter (H)</i>
DRRN (He et al., 2016b)	0.55	0.09	0.07	0.20	0.05	0.00
BYU-Agent (Fulda et al., 2017a)	0.59	0.03	0.00	0.10	0.00	0.01
Golovin (Kostka et al., 2017)	0.20	0.04	0.10	0.15	0.00	0.01
AE-DQN (Zahavy et al., 2018)	–	0.05	–	–	–	–
NeuroAgent (Rajalingam and Samothrakis, 2019)	0.19	0.03	0.00	0.20	0.00	0.00
NAIL (Hausknecht et al., 2019)	0.38	0.03	0.26	–	0.00	0.00
CNN-DQN (Yin and May, 2019a)	–	0.11	–	–	–	–
IK-OMP (Tessler et al., 2019)	–	1.00	–	–	–	–
TDQN (Hausknecht et al., 2020)	0.47	0.03	0.00	0.34	0.02	0.00
KG-A2C (Ammanabrolu and Hausknecht, 2020)	0.58	0.10	0.01	0.06	0.03	0.01
SC (Jain et al., 2020)	–	0.10	–	–	0.0	–
CALM (N-gram) (Yao et al., 2020)	0.79	0.07	0.00	0.09	0.00	0.00
CALM (GPT-2) (Yao et al., 2020)	0.80	0.09	0.07	0.14	0.05	0.01
RC-DQN (Guo et al., 2020a)	0.81	0.11	0.40	0.20	0.05	0.02
MPRC-DQN (Guo et al., 2020a)	0.88	0.11	0.52	0.20	0.05	0.02
SHA-KG (Xu et al., 2020)	0.86	0.10	0.10	–	0.05	0.02
MC!Q*BERT (Ammanabrolu et al., 2020b)	0.92	0.12	–	–	0.00	–
INV-DY (Yao et al., 2021)	0.81	0.12	0.06	0.11	0.05	–

Reinforcement Learning -> LLM

Most agents used to be **RL agents**, or a combination of RL + LLMs.

Now, most agents are **LLMs** with different strategies for:

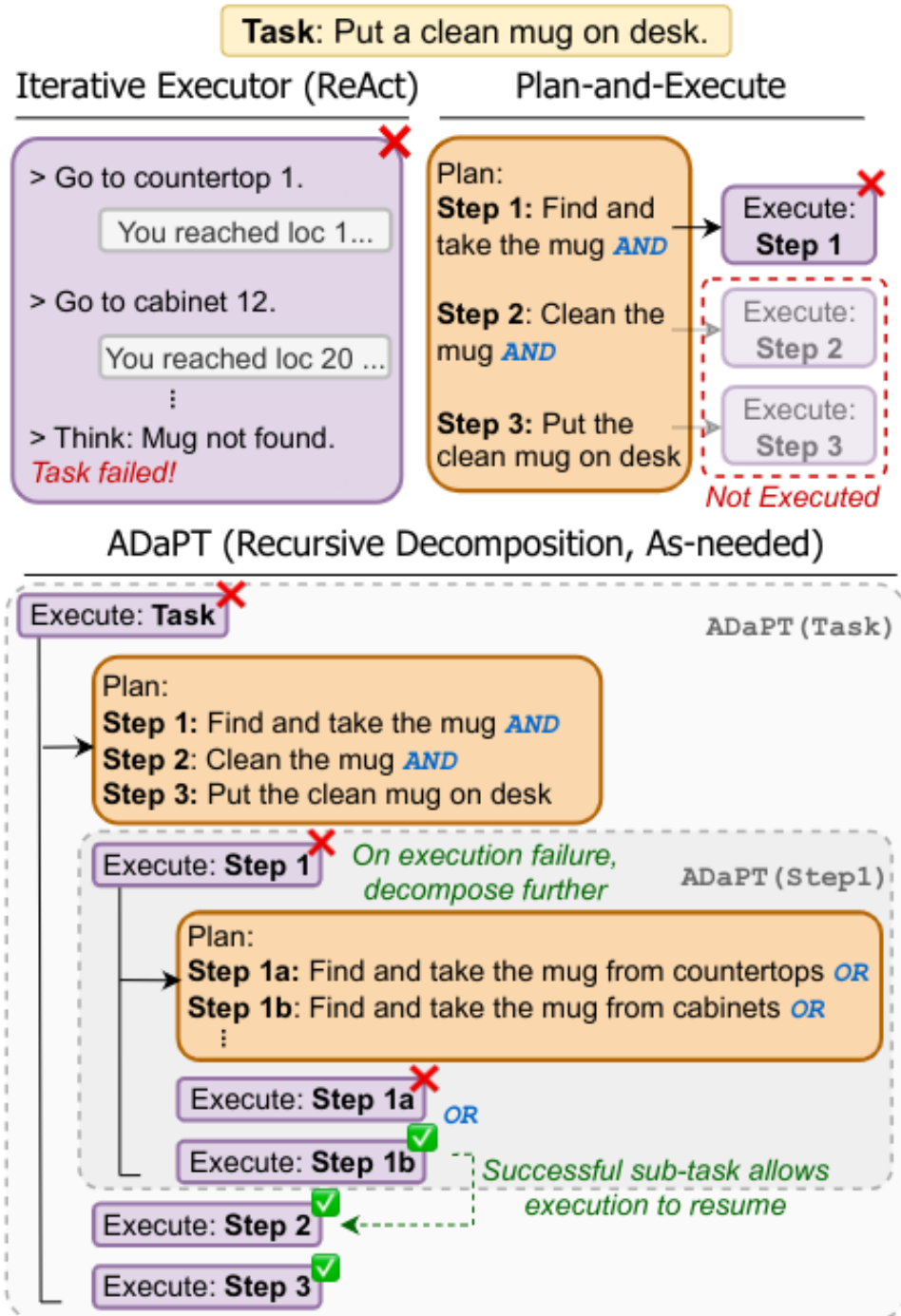
- Prompting
- Integrating external neurosymbolic modules (e.g. calculators, navigation modules)
- Memories
- Planners

Unintentional consequence:

- RL models: FAST, used to learn from 1 million+ steps
- Hard to use LLMs for tasks with large numbers of steps, or slow learning (since they tend to be slow).

ADAPT: As-Needed Decomposition and Planning with Language Models (Prasad et al., 2023)

- Iteratively decomposes larger tasks into smaller subtasks
- Generates plans for subtasks
- If the plan fails, it tries to recover and generate a new decomposition/plan
- Very large gains on ALFWorld, Webshop, and TextCraft.

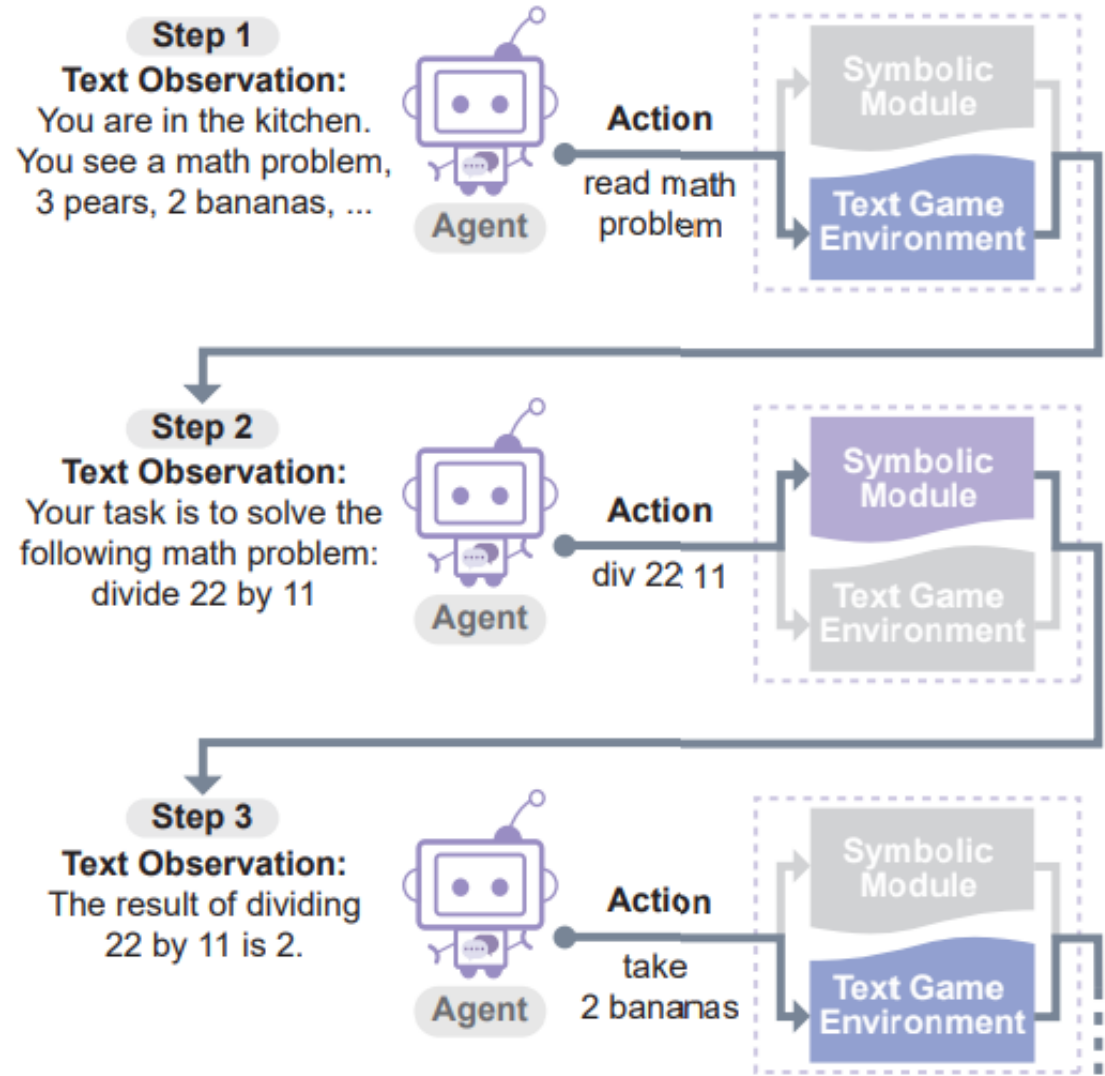


Behavior Cloned Transformers are Neurosymbolic Reasoners

(Wang et al., 2023)

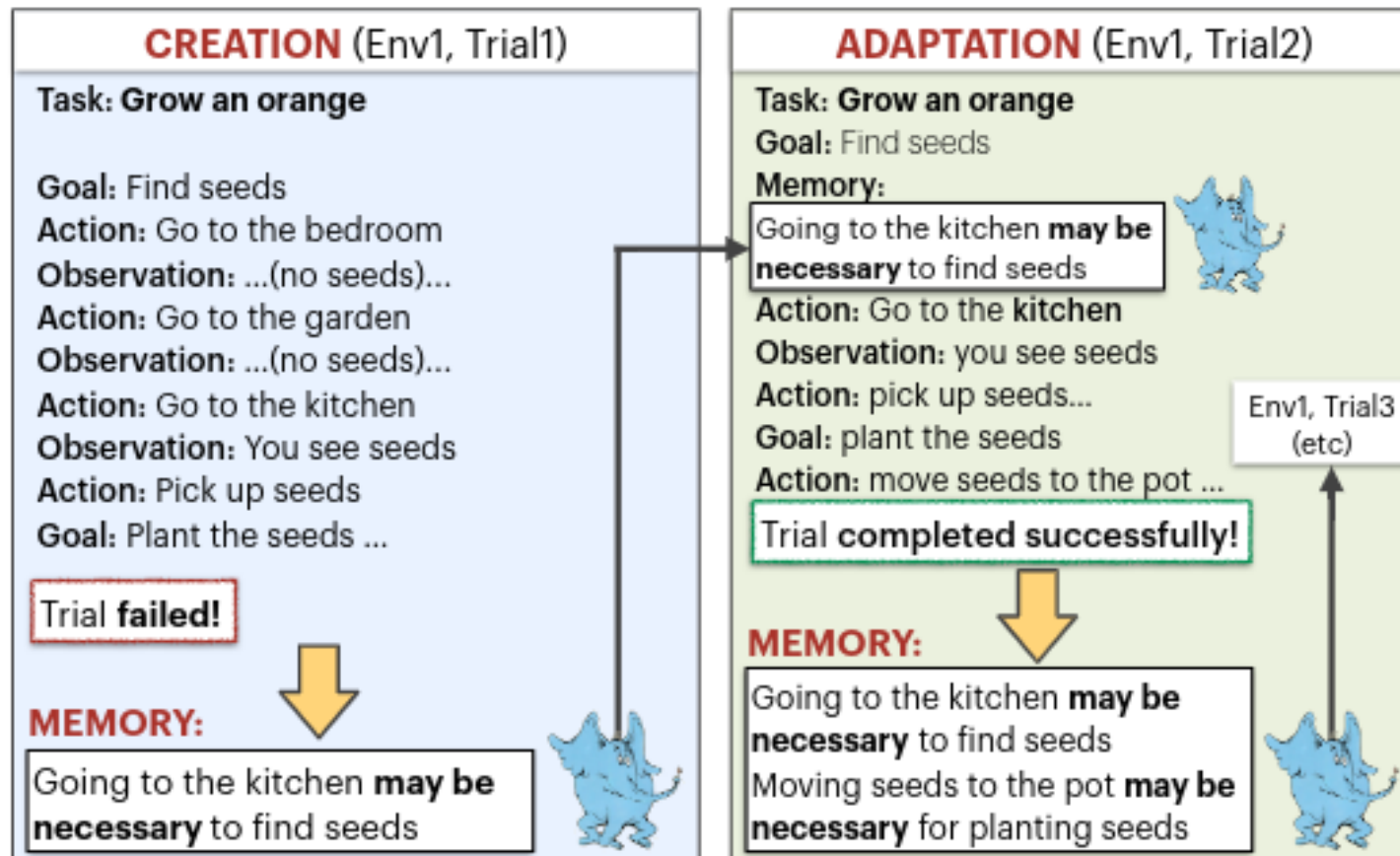
- Shows that you can easily hook up LLMs to external symbolic modules, like calculators, GPSes, knowledge base lookups, and other arbitrary Python code.
- Large performance gains: Completely solves 4 games that require these skills (including TWC).
- Also much more efficient solutions

Task Description: Your task is to solve the math problem. Then, pick up the item with the same quantity as the math problem answer, and place it in the box.



CLIN for Rapid Task Adaptation (Majumder et al., 2023)

- Agent tries a game, reflects on its performance, then builds a memory of “lessons” that it can use the next time it tries.
- Can learn many ScienceWorld tasks quickly, in a continual-learning paradigm



Remember what you did so you know what to do next (Ciosici et al., 2023)

- The prompts of most agents look something like this:
 1. A description of the task (“You need to cook this recipe...”)
 2. A description of the action space (“actions: take, put, slice, cook, ...”)
 3. A description of what the agent sees right now:

“You’re in the kitchen. You see a fridge, a stove, and a cupboard containing...”
 4. History: The last thing the agent saw, and last action it took.
- “More history is all you need”
 - If your LLM has enough context to fit in more steps of history, then it will do better.

Open Areas of Agent Research

- **Long-horizon tasks:** For problems that require many (dozens++) of steps to solve can be hard/expensive for LLMs.
- **Neurosymbolic Reasoning:** Augmenting an LLM with a new ability (like navigation) by giving it access to some Python code is popular
 - Easy stuff is starting to get picked over
 - Evergreen: There are always more abilities to augment LLMs with
- **Granularity Alignment:** The step-by-step plan that an LLM generates will often be at too high/low level for a given text game.
 - LLM: “Put some ice in the glass”
 - Environment: “go kitchen, open freeze, take ice, close freezer, go patio, put ice in glass”